At the time of the Action:

    Pending Claims: 1-4, 6-13, 15-30, 32-35, 37-44, and 46-49

    Withdrawn Claims: 50-52

    Canceled Claims: 5, 14, 31, 36, and 45

After this Response:

    Pending Claims: 1-4, 6-13, 15-30, 32-35, 37-44, and 46-49

    Amended Claims: 1, 2, 8, 11, 13, 15, 18, 20, 22, 23, 26-30, 32, 33, 37, 39, 42, 46, and 47

    Withdrawn: 50-52

    Canceled Claims: 5, 14, 31, 36, and 45

    New Claims: None

1.   (Currently Amended) A method, comprising:

requesting data to be streamed from a source device to a client device over a network;

building, by a parent Distributed Media Session ("DMS"), a distributed software infrastructure from an optimized distributed topology, the built distributed software infrastructure configured to stream data to the client device from the source device without rendering the data by the source device, the parent DMS providing a federated mechanism for control; [[and]]

resolving, by the parent DMS, a distributed topology from the request, wherein:

the distributed topology references a plurality of software components that, when executed, fulfill the request; [[and]]

at least one of the plurality of software components is executable on each of:

the source device; and

the client device[[.]],

instantiating, by the parent DMS, one or more child DMS; and

delegating control of one or more of the plurality of software components of the distributed software infrastructure to the one or more child DMS.

2.　　(Currently Amended) A method as described in claim 1, wherein the resolving further comprises:

discovering the capabilities of the client device to render a stream of data;

discovering the capabilities of the source device to stream data that is to be rendered, the discovering the capabilities of the client device and the source device by querying a look-up table that specifies a particular capability; and

deriving the distributed topology from both [[said]] the client device capabilities and the source device capabilities.

3.　　(Original) A method as described in claim 1, wherein the distributed topology is selected from the group consisting of:

a remote sink distributed topology;

a remote source distributed topology; and

a third party distributed topology.

4.   (Original) A method as described in claim 1, further comprising building a distributed software infrastructure from the distributed topology, wherein the distributed software infrastructure includes the plurality of software components.

5.   (Canceled).

6.   (Original) A method as described in claim 1, wherein:

the request also requests streaming data from an additional source device to the client device; and

the resolving resolves the distributed topology such that the plurality of software components, when executed, fulfills the request to stream data from each of the source device and the additional source device, respectively, to the client device.

7.   (Original) A method as described in claim 1, wherein:

the request also requests streaming data from the source device to an additional client device; and

the resolving resolves the distributed topology such that the plurality of software components, when executed, fulfills the request to stream data from the source device to each of the client device and the additional client device.

8.   (Currently Amended) A method as described in claim 1, wherein the distributed software infrastructure includes [[a]] the parent DMS ~~distributed media session that provides a federated mechanism for control~~, whereby:

the at least one software component that is executable on the source device is controllable by the ~~distributed media session~~ parent DMS; and

the at least one software component that is executable on the client device is controllable by the ~~distributed media session~~ parent DMS.

9.   (Original) A method as described in claim 1, wherein the resolving is executed without user intervention on a device selected from the group consisting of:

the source device;

the client device; and

a third party device.

10.   (Previously Presented) One or more computer-readable storage media comprising computer-executable instructions that, when executed, perform the method as recited in claim 1.

11.   (Currently Amended) A method, comprising:

receiving a request to stream data from a source device to a client device over a network; and

resolving, by a parent Distributed Media Session ("DMS"), a distributed topology

that references software components to fulfill the request, the parent DMS providing a federated mechanism for control, wherein the distributed topology is resolved from:

capabilities of the client device to render a stream of data; and

capabilities of the source device to stream data that is to be rendered;

[[and]]

building, by the parent DMS, a distributed software infrastructure from an optimized distributed topology, the built distributed software infrastructure configured to stream data to the client device from the source device without rendering the data by the source device, and building from the distributed topology a distributed software infrastructure that includes the referenced software components, wherein at least one of the software components is executable on each of:

the source device; and

the client device[[.]], and

instantiating, by the parent DMS, one or more child DMS; and

delegating control of the referenced software components included within the distributed software infrastructure to the one or more child DMS.

12.   (Original) A method as described in claim 11, wherein the distributed topology is selected from the group consisting of:

a remote sink distributed topology;

a remote source distributed topology; and

a third party distributed topology.

13.   (Currently Amended) A method as described in claim 11, wherein the resolving further comprises:

discovering the capabilities of the client device to render a stream of data;

discovering the capabilities of the source device to stream data that is to be rendered, the discovering the capabilities of the client device and the source device by querying a look-up table that specifies a particular capability; and

deriving a distributed topology from both said capabilities the capabilities of the client device and the capabilities of the source device, wherein the distributed topology references the software components.

14.   (Canceled).

15.   (Currently Amended) A method as described in claim 11, wherein the distributed topology references a distributed media session that provides a federated mechanism for control the parent DMS such that:

the at least one software component that is executable on the source device is controllable by the distributed media session parent DMS; and

the at least one software component that is executable on the client device is controllable by the distributed media session parent DMS.

16.   (Original) A method as described in claim 11, wherein the receiving and the resolving are executed without user intervention on a device selected from the group

consisting of:

> the source device;
>
> the client device; and
>
> a third party device.

17. (Original) One or more computer-readable media comprising computer-executable instructions that, when executed, perform the method as recited in claim 11.

18. (Currently Amended) A method, comprising:

> discovering the capabilities of a client device to render a stream of data;
>
> discovering the capabilities of a source device to stream data that is to be rendered, the discovering the capabilities of the client device and the source device by querying a look-up table that specifies a particular capability;
>
> building, by a parent Distributed Media Session ("DMS"), a distributed software infrastructure from an optimized distributed topology, the built distributed software infrastructure configured to stream data to the client device from the source device without rendering the data by the source device, the parent DMS providing a federated mechanism for control; [[and]]
>
> deriving, by the parent DMS, a distributed topology from both said capabilities the capabilities of the client device and the capabilities of the source device, wherein:
>
> > the distributed topology references a plurality of software components to fulfill the request; and

at least one of the software components referenced by the distributed topology is executable on each of:

> the source device; and

> the client device[[.]], and

> instantiating, by the parent DMS, one or more child DMS; and

> delegating control of one or more of the referenced plurality of software components to the one or more child DMS.

19. (Original) A method as described in claim 18, wherein the distributed topology is selected from the group consisting of:

a remote sink distributed topology;

a remote source distributed topology; and

a third party distributed topology.

20. (Currently Amended) A method as described in claim 18, further comprising building from the distributed topology a distributed software infrastructure that includes [[said]] the software components.

21. (Original) A method as described in claim 18, wherein:

the discovering of the capabilities of the client device further comprises examining the client device to find a software component which renders a stream of data; and

the discovering of the capabilities of the source device further comprises

examining the source device to find a software component which streams data.

22.   (Currently Amended) A method as described in claim 18, wherein the ~~discovering of the capabilities of the client and source devices, respectively, further comprises querying a~~ look-up table [[that]] contains:

the capabilities of the client device to render the stream of data; and

the capabilities of the source device to stream data that is to be rendered.

23.   (Currently Amended) A method as described in claim 18, wherein the distributed topology references ~~a distributed media session~~ the parent DMS that provides a federated mechanism for control such that:

the at least one software component that is executable on the source device is controllable by the ~~distributed media session~~ parent DMS; and

the at least one software component that is executable on the client device is controllable by the ~~distributed media session~~ parent DMS.

24.   (Original) A method as described in claim 18, wherein the receiving and the resolving are executed without user intervention on a device selected from the group consisting of:

the source device;

the client device; and

a third party device.

25.   (Original) One or more computer-readable media comprising computer-executable instructions that, when executed, perform the method as recited in claim 18.

26.   (Currently Amended) A method, comprising:

receiving a request to stream data from a source device to a client device;

discovering the capabilities of the client device to render a stream of data;

discovering the capabilities of the source device to stream data that is to be rendered, the discovering the capabilities of the client device and the source device by querying a look-up table that specifies a particular capability;

deriving, by a parent Distributed Media Session ("DMS"), a distributed topology to fulfill the request from both said the capabilities of the client device and the capabilities of the source device, wherein the distributed topology references a plurality of software components, the parent DMS providing a federated mechanism for control;

building, by the parent DMS, from the distributed topology a distributed software infrastructure, wherein the distributed software infrastructure includes [[said]] the software components referenced by the distributed topology;

building, by the parent DMS, a distributed software infrastructure from an optimized distributed topology, the built distributed software infrastructure configured to stream data to the client device from the source device to the client device without rendering the data by the source device;

instantiating, by the parent DMS, one or more child DMS;

delegating control of one or more of the plurality of software components

referenced by the distributed topology to the one or more child DMS;

streaming the data from the source device to the client device; and

rendering the data by the client device.

27.   (Currently Amended) A parent distributed media session ("DMS"), comprising

a software component residing on a storage medium having instructions that, when executed, directs acts comprising:

resolving, by the DMS that provides a federated mechanism for control, a distributed topology that references a plurality of software components that, when executed, fulfill a request to stream data from a source device to a client device; wherein the resolving further comprises optimizing the distributed topology such that the distributed software infrastructure which is built from the distributed topology is configured to stream data from the source device to the client device without rendering the data by the source device before the data is streamed; [[and]]

building, from the distributed topology, a distributed software infrastructure that includes [[said]] the software components, wherein at least one of the [[said]] software components is executable on each of:

the source device; and

the client device[[.]],

instantiating, by the parent DMS, one or more child DMS; and

delegating control of one or more of the plurality of software components to the

one or more child DMS.

28.   (Currently Amended) ~~A distributed media session~~ The parent DMS as described in claim 27, wherein the resolving further comprises:

discovering the capabilities of the client device to render a stream of data;

discovering the capabilities of the source device to stream data that is to be rendered, the discovering the capabilities of the client device and the source device by querying a look-up table that specifies a particular capability; and

deriving the distributed topology from ~~both said~~ t he ca pabilities of the client device and the capabilities of the source device.

29.   (Currently Amended) ~~A distributed media session~~ The parent DMS as described in claim 27, wherein the distributed topology is selected from the group consisting of:

a remote sink distributed topology;

a remote source distributed topology; and

a third party distributed topology.

30.   (Currently Amended) ~~A distributed media session~~ The parent DMS as described in claim 27, wherein the building further comprises supplying at least one additional software component which is referenced by the distributed topology.

31.   (Canceled).

32.   (Currently Amended) A computer-readable medium comprising computer-executable instructions residing on a storage medium that, when executed, direct a computing device to perform acts comprising:

resolving, by a parent Distributed Media Session ("DMS") and without user intervention, a distributed topology that references a plurality of software components that, when executed, stream data from a source device to a client device over a network;

building, by the parent DMS, a distributed software infrastructure from an optimized distributed topology, the built distributed software infrastructure configured to stream data to the client device from the source device to the client device without rendering the data by the source device; [[and]] wherein at least one of the plurality of software components is executable on each of:

the source device; and

the client device[[.]],

instantiating, by the parent DMS, one or more child DMS; and

delegating control of one or more of the plurality of software components to the one or more child DMS.

33.   (Currently Amended) A computer-readable medium as described in claim 32, wherein the resolving further comprises:

discovering the capabilities of the client device to render a stream of data;

discovering the capabilities of the source device to stream data that is to be rendered, the discovering the capabilities of the client device and the source device by querying a look-up table that specifies a particular capability; and

deriving the distributed topology from both said t he ca pabilities of the client device and the capabilities of the source device.

34.   (Original) A computer-readable medium as described in claim 32, wherein the distributed topology is selected from the group consisting of:

a remote sink distributed topology;

a remote source distributed topology; and

a third party distributed topology.

35.   (Original) A computer-readable medium as described in claim 32, further comprising building a distributed software infrastructure from the distributed topology.

36.   (Canceled).

37.   (Currently Amended) A computer-readable storage medium comprising computer-executable instructions that, when executed, direct a computing device to perform acts comprising:

discovering the capabilities of a client device to render a stream of data;

discovering the capabilities of a source device to stream data that is to be

rendered, the discovering the capabilities of the client device and the source device by querying a look-up table that specifies a particular capability; [[and]]

deriving, by a parent Distributed Media Session ("DMS") and without user intervention, a distributed topology from both [[said]] the capabilities of the client device and the capabilities of the source device, wherein:

the distributed topology references a plurality of software components that, when executed, stream data, without rendering the data, from the source device to the client device; and

at least one of the plurality of software components referenced by the distributed topology is executable on each of:

the source device; and

the client device[[.]],

instantiating, by the parent DMS, one or more child DMS; and

delegating control of one or more of the plurality of software components to the one or more child DMS.

38.   (Original) A computer-readable medium as described in claim 37, further comprising building from the distributed topology a distributed software infrastructure that includes the plurality of software components.

39.   (Currently Amended) A system, comprising:

a source device that is operable to stream data to be rendered;

a client device that is operable to render a stream of data; and

a _parent_ distributed media session ("DMS"), which when executed, causes actions to be performed including:

resolving, by the parent DMS, a distributed topology that references a plurality of software components that, when executed, stream data from the source device to the client device over a network; [[and]]

building, by the parent DMS, from the distributed topology a distributed software infrastructure that includes [[said]] the software components, wherein the building further comprises building the distributed software infrastructure from an optimized distributed topology, the built distributed software infrastructure configured to stream data to the client device from the source device without rendering the data by the source device, wherein at least one of the [[said]] software components is executable on each of:

the source device; and

the client device[[.]],

instantiating one or more child DMS; and

delegating control of one or more of the plurality of software components to the one or more child DMS.

40.   (Original) A system as described in claim 39, wherein the source device is selected from the group consisting of:

a computing device which is locally connected to a source peripheral device; and

a network-ready device that is operable to stream data that is to be rendered.

41.   (Original) A system as described in claim 39, wherein the client device is selected from the group consisting of:

a computing device which is locally connected to a rendering device; and

a network-ready device suitable for rendering data.

42.   (Currently Amended) A system as described in claim 39, wherein the resolving further comprises:

discovering the capabilities of the client device to render a stream of data;

discovering the capabilities of the source device to stream data that is to be rendered, the discovering the capabilities of the client device and the source device by querying a look-up table that specifies a particular capability; and

deriving the distributed topology from both said t he ca pabilities of the client device and the capabilities of the source device.

43.   (Original) A system as described in claim 39, wherein the distributed topology is selected from the group consisting of:

a remote sink distributed topology;

a remote source distributed topology; and

a third party distributed topology.

44.   (Original) A system as described in claim 39, wherein the building further comprises supplying at least one software component that is referenced by the

distributed topology.

45.   (Canceled).

46.   (Currently Amended) A system as described in claim 39, wherein the execution of the ~~distributed media session~~ parent DMS is performed by one of:

the source device;

the client device; and

a third party device.

47.   (Currently Amended) A system, comprising:

a source device which includes a software component that, when executed by the source device, streams data that is to be rendered;

a client device which includes a software component that, when executed by the client device, renders a stream of data; and

a parent distributed media session ("DMS"), which when executed by either the source device or the client device, provides a federated mechanism for control of:

the software component that, when executed by the source device, streams data without rendering the data that is to be rendered; and

the software component that, when executed by the client device, renders a stream of data[[.]],

the parent DMS instantiating one or more child DMS and delegating control of the

software component to the one or more child DMS.

48. (Original) A system as described in claim 47, wherein the source device is selected from the group consisting of:

a computing device which is locally connected to a source peripheral device; and

a network-ready device that is operable to stream data that is to be rendered.

49. (Original) A system as described in claim 47, wherein the client device is selected from the group consisting of:

a computing device which is locally connected to a rendering device; and

a network-ready device suitable for rendering data.

50. (Withdrawn) A system comprising:

a network;

a source device which is configured to:

compress data; and

stream the compressed data without rendering the compressed data; and

a client device, communicatively coupled to the source device over the network, wherein the client device is configured to:

receive the streamed data from the source device over the network;

decompress the received data; and

render the decompressed data.

51.   (Withdrawn) A system as described in claim 50, wherein the source device is selected from the group consisting of:

a computing device which is locally connected to a source peripheral device; and

a network-ready device that is operable to stream data that is to be rendered.

52.   (Withdrawn) A system as described in claim 50, wherein the client device is selected from the group consisting of:

a computing device which is locally connected to a rendering device; and

a network-ready device suitable for rendering data.